

REMARKS

Reconsideration of the application in view of the following remarks is respectfully requested. No claims have been canceled or amended. Claims 37-99 are currently pending in the application.

As an initial matter, Applicant notes that claims 53, 54, 74, 75, 95, and 96 were not explicitly rejected in the Final Office Action. Thus, Applicant assumes that these claims are allowable.

In the Final Office Action, the Examiner rejected claims 37-39, 44-45, 48, 51-52, 55-60, 65-66, 69, 72-73, 76-81, 86-87, 90, 93-94, and 97-99 under 35 U.S.C. §102(e) as being anticipated by Erlenkoetter et al. (U.S. Patent No. 6,253,254 B1, hereinafter, "Erlen"). This rejection is respectfully traversed.

Independent Claim 37

Independent method claim 37 recites:

A method implemented by a server, comprising:
receiving a request from a first client to browse contents of a first file system on a first data server, wherein the first data server implements the first file system for managing file access and storage, and wherein the first client is unaware that the first data server implements the first file system;
selecting a first protocol interpreter from a plurality of different protocol interpreters, wherein the first protocol interpreter implements a first file access protocol which enables interaction with the first file system;
invoking the first protocol interpreter to interact with the first file system of the first data sever to obtain therefrom a first list of contents, wherein the first list of contents sets forth a hierarchical listing of at least a portion of the contents of the first file system on the first data server, the first list of contents comprising one or more directories and zero or more files; and
sending at least a portion of the first list of contents to the first client. (emphasis added)

The method of claim 37 provides an advantageous way for a client to interact with the file system of a data server without being aware of the file system that is implemented on the data server. According to claim 37, this is achieved via a server (referred to in the following discussion as the intermediate server). Specifically, when the intermediate server receives a request from a client to browse the contents of a file system on a data server, the intermediate server selects a protocol interpreter from a plurality of different protocol interpreters. The selected protocol interpreter implements a file access protocol, which enables the intermediate server to interact with the file system on the data server.

Once the protocol interpreter is selected, the intermediate server invokes the protocol interpreter to interact with the file system on the data server. This interaction enables the intermediate server to obtain from the data server a list of contents. This list of contents sets forth a hierarchical listing of at least a portion of the contents of the file system. This list of contents comprises one or more directories and zero or more files. The intermediate server then provides the list of contents to the client. By doing so, the intermediate server in effect provides the client with a view of a portion of the contents of the file system of the data server, thereby allowing the client to browse the file system. This is achieved without the client even being aware of the file system that is implemented on the data server.

Also, because the intermediate server selects the protocol interpreter from a plurality of different protocol interpreters, the intermediate server can, by selecting the proper protocol interpreters, interact with different data servers implementing different file systems. This in turn means that the client, via the intermediate server, can browse the file system contents of different data servers implementing different file systems. Thus, without even being aware of any file system implemented by any data servers, the

client is able to browse the file system contents of a number of different data servers implementing a number of different file systems.

Such a method is neither disclosed nor suggested by Erlen. Instead, Erlen discloses a method and apparatus for managing distributed objects. In Erlen, a server receives a hypermedia request from a client. The server converts the hypermedia request into an object request, and sends the object request to an object manager. The object manager retrieves information pertaining to the requested object, and sends a response to the server. In turn, the server converts the response into a hypermedia format that the client can understand, and sends the converted response to the client. In this manner, the client is able to obtain information pertaining to the requested object. (See Col. 2, lines 21-31)

The method of claim 37 is quite different from the method of Erlen, and in no way does Erlen disclose or suggest the method of claim 37. To illustrate this, Applicants would like to note at least one very significant point: the method of claim 37 is directed to the browsing of the contents of a file system, whereas the method of Erlen is directed to the management of objects in an object management system. In the computer science art, file systems and object management systems are fundamentally different, and one cannot be held to be analogous to the other. First of all, the entities that they manage are fundamentally different. A file is akin to a physical entity that is composed of a group or a set of bits. A file may be accessed as a whole, deleted as a whole, copied, and moved from directory to directory. To access, delete, copy, move, etc. a file, it is not necessary to understand the contents of the file. In contrast, an object is a logical entity. An object is recognized only by an object manager that recognizes the data structure of the object. An object is not a generally recognized entity that can be accessed, deleted, copied, or

moved from directory to directory. For example, unlike a file, it is not possible to store a copy of just an object on a floppy disk. To drive home the difference between a file and an object, reference will be made to the example of an object oriented database. In such a database, the entire database is embodied as a file. Inside the file are bits that represent data structures that make up one or more objects. These data structures are recognized by a database management system as one or more objects. Thus, as this example shows, an object is a logical entity that is recognized by an object management system, while a file is the physical entity that encompasses the bits that make up the data structures that represent one or more objects.

In addition to managing very different entities, files systems and object management systems use very different techniques and structures to manage their respective entities. As is well known, a file system uses directories to group and to organize files. Each directory can contain one or more directories (referred to as subdirectories) and/or files, and each subdirectory can contain one or more subdirectories and/or files. Because a directory can contain one or more subdirectories, and because each subdirectory can contain one or more subdirectories, a hierarchy of directories can result. In contrast, there is no such notion as a directory in an object management system. In an object management system, objects are not grouped under directories (e.g. there is no notion of objects A and B being under directory 1 and objects C and D being under directory 2). Rather, they are grouped by association (e.g. object A has a link to and hence is associated with object B). Because of this, all objects are at the same level, which means that there is no hierarchy. The properties (e.g. components, attributes, etc.) of each object may be set forth in a hierarchical manner (as shown in Fig. 4 of Erlen), but the objects themselves are not grouped and organized in a hierarchical manner. As this

discussion shows, a file system manages files in a fundamentally different way than an object management system manages objects. Because they manage fundamentally different entities in fundamentally different ways, file systems and object management systems cannot be held to be analogous.

In claim 37, it is specifically recited that the server receives "a request from a first client to browse contents of a first file system on a first data server". In Erlen, the request received from the client is a request to obtain information pertaining to an object in an object management system. As argued above, a file system is fundamentally different from an object management system, and a request to browse a file system is fundamentally different from a request for information pertaining to an object. There is nothing in Erlen that discloses or suggests that the server receives a request to browse the contents of a file system. Consequently, this limitation of claim 37 is not met by Erlen.

Also, claim 37 specifically recites that the first list of contents obtained from the data server comprises "one or more directories". As argued above, an object management system does not have the notion of a directory; thus, in Erlen, the response received by the server from the object manager cannot comprise a directory. In asserting that this limitation is disclosed in Erlen, the Examiner cites Col. 5, lines 42-46, 53-55, Col. 6, lines 24-37, and Col. 7, lines 40-60. From these excerpts, it appears that the Examiner is interpreting an object, as that term is used in Erlen, as a directory. This interpretation cannot hold. As noted previously, a directory in a file system can comprise one or more other directories. The objects in Erlen do not have this property. In Erlen, there is no mention whatsoever that an object can contain another object. An object can reference another object, and an object can contain components and attributes, but an object cannot

contain another object. Thus, the objects in Erlen cannot in any way be interpreted as directories. For at least this reason, this limitation of claim 37 is not met by Erlen.

Furthermore, claim 37 specifically recites that the server selects "a first protocol interpreter from a plurality of different protocol interpreters". There is no such teaching in Erlen. As far as Applicant can see, the server of Erlen uses only one translator to interact with the object manager. It does not select among a plurality of different translators. Thus, even if the translator of Erlen could be interpreted as a protocol interpreter (and it probably could not because the translator does not implement a file access protocol as required by claim 37), there is no teaching of selecting the translator from a plurality of different translators. Hence, this limitation of claim 37 is not disclosed or suggested by Erlen.

As argued above, Erlen fails to disclose or suggest several limitations of claim 37. For at least this reason, Applicant submits that claim 37 is patentable over Erlen.

Claims Depending from Claim 37

Claims 38-39, 44-45, 48, 51-52, and 55-57 depend from, and hence, incorporate all of the limitations of claim 37. Thus, Applicant submits that these claims are likewise patentable over Erlen for at least the reasons given above in connection with claim 37.

In addition to being dependent from claim 37, these claims recite further limitations that render them patentable over Erlen. For example, claim 45 recites "receiving a request...to access a particular file", "invoking the first protocol interpreter...to retrieve the particular file", "determining a file type for the particular file", "generating a set of encoding information...wherein the encoding information comprises information for causing the first client to execute a particular type of application to

process the particular file", and "sending the particular file and the set of encoding information to the first client". In rejecting this claim, the Examiner cites Col. 6, lines 35-45 and Col. 8, lines 50-56 of Erlen. These excerpts, however, say nothing about determining a file type, generating a set of encoding information, or sending the particular file and the set of encoding information to the client. These limitations of the claim are not disclosed or suggested by Erlen. Accordingly, Applicant submits that claim 45 is patentable over Erlen.

Also, claim 51 recites "receiving a request...to compress a particular file", "invoking the first protocol interpreter...to retrieve the particular file", and "compressing the particular file". In rejecting this claim, the Examiner cites Col. 8, lines 8-12 and Col. 9, lines 20-28 of Erlen. These excerpts, however, say nothing about receiving a request to compress a file, or actually compressing the file. These limitations of the claim are not disclosed or suggested by Erlen. Accordingly, Applicant submits that claim 51 is patentable over Erlen.

Also, claim 52 recites "receiving a request...to send a particular file...to a recipient", "invoking the first protocol interpreter...to retrieve the particular file", and "sending the particular file to the recipient without first downloading the particular file to the first client". In rejecting this claim, the Examiner cites Col. 8, lines 8-12 and Col. 8, lines 21-30 of Erlen. These excerpts, however, say nothing about receiving a request to send a particular file to a recipient, or actually sending the particular file to the recipient without first downloading the particular file to the first client. These limitations of the claim are not disclosed or suggested by Erlen. Accordingly, Applicant submits that claim 52 is patentable over Erlen.

Claims 58-60, 65-66, 69, 72-73, and 76-78

Claims 58-60, 65-66, 69, 72-73, and 76-78 are apparatus claims, which are analogous to the method claims of claims 37-39, 44-45, 48, 51-52, and 55-57. Applicant submits that claims 58-60, 65-66, 69, 72-73, and 76-78 are patentable over Erlen for at least the same reasons as those given above in connection with claims 37-39, 44-45, 48, 51-52, and 55-57.

Claims 79-81, 86-87, 90, 93-94, and 97-99

Claims 79-81, 86-87, 90, 93-94, and 97-99 are computer readable medium claims which are analogous to the method claims of claims 37-39, 44-45, 48, 51-52, and 55-57. Applicant submits that claims 79-81, 86-87, 90, 93-94, and 97-99 are patentable over Erlen for at least the same reasons as those given above in connection with claims 37-39, 44-45, 48, 51-52, and 55-57.

Claims 40-43, 61-64, and 82-85

In the Final Office Action, the Examiner rejected claims 40-43, 61-64, and 82-85 under 35 U.S.C. §103(a) as being unpatentable over Erlen in view of Stollfus et al. (U.S. Patent No. 6,321,258). This rejection is respectfully traversed.

Dependent claims 40-43 depend from, and hence, incorporate all of the limitations of claim 37. If claim 37 is patentable over Erlin and Stollfus, then it follows that claims 40-43 are also patentable over Erlin and Stollfus.

As argued above, Erlin fails to disclose or suggest several limitations of claim 37. These limitations are also not disclosed or suggested by Stollfus (and the Examiner has not contended that these limitations are shown by Stollfus). Thus, even if Erlen and

Stollfus are combined (assuming arguendo that it would have been obvious to combine the references), the combination still would not give rise to the invention claimed in claim 37. Thus, Applicant submits that claim 37 is patentable over Erlin and Stollfus, taken individually or in combination. Applicant further submits that claims 40-43, which depend from claim 37, are likewise patentable over Erlin and Stollfus for at least the same reasons as those given above in connection with claim 37.

Claims 61-64 are apparatus claims, which are analogous to the method claims of claims 40-43. Applicant submits that claims 61-64 are patentable over Erlin and Stollfus for at least the same reasons as those given above in connection with claims 40-43.

Claims 82-85 are computer readable medium claims, which are analogous to the method claims of claims 40-43. Applicant submits that claims 82-85 are patentable over Erlin and Stollfus for at least the same reasons as those given above in connection with claims 40-43.

Claims 46-47, 49-50, 67-68, 70-71, 88-89, and 91-92

In the Final Office Action, the Examiner rejected claims 46-47, 49-50, 67-68, 70-71, 88-89, and 91-92 under 35 U.S.C. §103(a) as being unpatentable over Erlin in view of Busey et al. (U.S. Patent No. 5,764,916). This rejection is respectfully traversed.

Dependent claims 46-47 and 49-50 depend from, and hence, incorporate all of the limitations of claim 37. If claim 37 is patentable over Erlin and Busey, then it follows that claims 46-47 and 49-50 are also patentable over Erlin and Busey.

As argued above, Erlin fails to disclose or suggest several limitations of claim 37. These limitations are also not disclosed or suggested by Busey (and the Examiner has not contended that these limitations are shown by Busey). Thus, even if Erlin and Busey are combined (assuming *arguendo* that it would have been obvious to combine the references), the combination still would not give rise to the invention claimed in claim 37. Thus, Applicant submits that claim 37 is patentable over Erlin and Busey, taken individually or in combination. Applicant further submits that claims 46-47 and 49-50, which depend from claim 37, are likewise patentable over Erlin and Busey for at least the same reasons as those given above in connection with claim 37.

Claims 67-68 and 70-71 are apparatus claims, which are analogous to the method claims of claims 46-47 and 49-50. Applicant submits that claims 67-68 and 70-71 are patentable over Erlin and Busey for at least the same reasons as those given above in connection with claims 46-47 and 49-50.

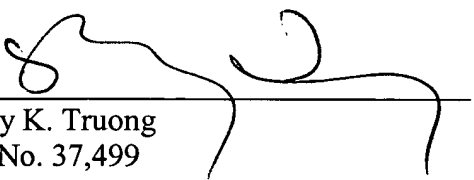
Claims 88-89 and 91-92 are computer readable medium claims, which are analogous to the method claims of claims 46-47 and 49-50. Applicant submits that claims 88-89 and 91-92 are patentable over Erlin and Busey for at least the same reasons as those given above in connection with claims 46-47 and 49-50.

For the reasons set forth above, Applicant respectfully submits that claims 37-99 are patentable over the art of record, including the art cited but not applied. Accordingly, allowance of all pending claims is hereby respectfully solicited.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Dated: September 26, 2005


Bobby K. Truong
Reg. No. 37,499

2055 Gateway Place, Suite 550
San Jose, California 95110-1089
Telephone No.: (408) 414-1080
Facsimile No.: (408) 414-1076

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

on 9-26-05 by 